Reading Guttenberg

February 14

2008

Project Guttenberg is a large, free archive of public domain texts. They offer audio-books as well, and are always in need of more. A MapReduce system could be designed to leverage the power of distributed computing to create audio books using existing text-to-speech solutions.

Overview

Project Guttenberg is the web's oldest archive of public domain texts, originally begun with the aim of making the 600 most-referenced books freely available to everyone. The collection now spans over twenty thousand volumes and grows daily. Project Guttenberg offers free downloads of any of the works in the archive, and are constantly working to improve the size and quality of the collection. Project Guttenberg also offers audio books in MP3 and other formats, for people who have difficulty reading printed versions or otherwise need to listen to the books rather than read them. Audio books are usually created by volunteers, either by reading the book aloud or using some form of text-to-speech technology. As I understand it only a fairly small subset of Project Guttenberg's printed collection is audible, so it would be fun to design a MapReduce system for a Hadoop cluster that could be used to generate audio books from text.

Suggested Solution and System Architecture

There are a number of free and open source text-to-speech programs, and some that are not free that tend to be of higher quality. It shouldn't be too difficult to take an open-source Java text-to-speech framework such as MARY (http://www.mary.dfki.de) and incorporate it into a MapReduce program. As experienced with the various Wikipedia projects we've done in the past, I expect that that the greatest difficulty of this project will be handling parsing (due to textual irregularities) and trying to get the best possible sound out of the audio books.

It might be good to develop a benchmark comparing the performance of the prototypical MapReduce text-to-speech program with known samples from Project Guttenberg. There will also be some amount of effort involved in trying to avoid redundant generation of books that are already in the archive of Project Guttenberg or one of its affiliates. Investigation will be needed to discover how best to shard the texts that come in, and might depend in some part on the API of whatever text-to-speech system is chosen.

Development Plan

1st week: More detailed feasibility study: Investigation of text-to-speech frameworks and APIs, licenses of derived works, etc. Getting an idea for the amount of computing required to generate speech from a fragment of text will help us to choose the scope of the project (number of books converted, etc.).

Remainder of time: Coding, execution of finished project on cluster.

Feasibility Rationale

Project Guttenberg and many of its affiliates have a large and free dataset that is not too difficult to acquire. No spidering is involved, the entire dataset can generally be downloaded automatically. Text-to-speech frameworks exist that do the complete job of taking a fragment of text and converting it to speech. The most fun/difficulty/profit in a project like this will come in figuring out how to fine-tune the system to improve the results.

Project Proposal Format:

You will submit an essay of no more than 2 pages of text (illustrations are free). Your essay should follow the outline below.

- **Overview** - 1-2 paragraphs.
  Describe and analyze the problem or idea, giving background on the problem and listing some of the properties of existing solutions (if the idea isn't new). Also, briefly explain your proposed solution, describing your top-level objectives, differentiators, and the scope of the work.
- **Suggested Solution and System Architecture** - 2-3 paragraphs.
  Describe your solution in more detail, including essential system features and organization. Provide an analysis of the technical feasibility at this level. If necessary, include a high-level sketch of the components and how they will integrate (illustrations may help).
- **Development Plan** - 1 paragraph.
  Describe a high-level timeline for this project, consisting of major milestones and their short descriptions (1-2 sentences). This should help you to scope the work and determine the number of developers you might need to complete your project.
- **Feasibility Rationale** - 1 paragraph.
  Evaluate the conceptual integrity of your idea and identify any risks. This is an appropriate space to list concerns upfront which might require additional help from instructors or staff.